# The Utility of Robot Sensory Devices in a Collaborative Autonomic Environment


Catherine Saunders, Roy Sterritt, George Wilkie

School of Computing and Mathematics

University of Ulster

Northern Ireland

saunders-c2@email.ulster.ac.uk, r.sterritt@ulster.ac.uk, fg.wilkie@ulster.ac.uk



***Abstract* —This paper proposes an Autonomic architecture that will enable mobile robots to self-manage and collaborate by using control loops to monitor their internal state and external environment. The Autonomic Computing MAPE-K control loop is used to design a Robot Autonomic Element and a Mapping Autonomic Element; each can exchange data and collaborate to find an object located within a room. A review of the sensor capabilities of the X80-H mobile robot platform is undertaken with emphasis on how useful each sensor will be to the proposed research. A literature review of other projects that feature robot collaboration is also included.**



***Keywords - autonomic computing; mobile robot; Dr Robot X80-H; collaboration; MAPE-K***


## I. INTRODUCTION

In 2001 IBM announced its Autonomic Computing [1] Initiative as a solution to the ever-growing complexity inherent in modern computer systems [2] [3]. Autonomic Computing seeks to solve the problems that occur when a system becomes too large to manage. System self-management is essential in order to reduce the amount of human involvement required [4].

The purpose of this research is to investigate how mobile robots could use Autonomic Computing concepts to collaborate and achieve a common goal. To collaborate effectively, robots need to be internally self-aware and aware of their external environment [5]. A swarm of mobile robots, with each robot representing an Autonomic Element (AE) makes up a larger Autonomic System. The relationship between these Autonomic Elements and how they interact and share information in order to collaborate is the focus of this research.

The research project will involve the design of an Autonomic system with multiple self-managing entities capable of exchanging data and collaborating with each other [5]. The aim is to have each robot operating within an enclosed environment and have them collaborate to find an object. During the search for the object, they must map their environment and relay meaningful information to each other.

In this paper we discuss the sensor capabilities of the Dr Robot X80-H platform that will be used for the research. A design of the proposed system is included and explained. A brief literature review of robotic collaborative systems is also included. The Research Background section looks at why self-managing software systems are needed.

## II. ROBOT SENSORY DEVICES

This section gives an overview of the sensors that the Dr Robot X80-H is equipped with. We assess how reliable they are and how useful they will be for our research. We will be using 4 Dr Robot X80-H mobile robots to demonstrate the autonomic architecture and software that will be developed for the research. The X80-H platform is a modified version of the X80 mobile robot; it features an X80 base but with an animated head. The head section has 2 eyes with a camera integrated into the right eye. The robot is a differential drive design and has two 7-inch wheels controlled by separate 12V motors. Its maximum speed is 1m per second but tests have proven that moving at this speed and then stopping abruptly can cause the robot to topple over.

The robot has a wireless antenna and is operated by sending instructions from a PC that is connected to a Dr Robot wireless router. Software cannot be downloaded directly to the robot; this is quite a typical setup for a mobile robot. The PC is able to send instructions to the robot and receive data back from the robot.

To develop an application, the WiRobot ActiveX Component must be added to Visual Studio. By creating an instance of this COM object on a Windows Form, it is then possible to use the Dr Robot SDK and instruct the robot. To send the information to the X80-H via the router, the WiRobot Gateway must also be used; it is a utility program running on the host PC and connects the PC to the X80-H by using the IP address of the robot.

### A. Ultrasonic

The X80-H's 3 ultrasonic sensors are positioned on the front of the base unit and can detect an object within the range of 4-255cm. If an object is closer than 4cm, the distance will be displayed as 4cm [6]. The Ultrasonic works by sending a sound wave and calculating how long it takes for the sensor to receive the sound wave after it bounces off an object. The Ultrasonic sensor consists of two round objects, one is a speaker that sends the wave and the other is a microphone that receives the echo. We have found the sensors to be very accurate and have used them for basic collision detection.

### B. Infrared

The X80-H is equipped with 8 Infrared (IR) sensors that are capable of detecting the distance to an object. The IR sensors can detect an object when it is within the range of 8-80cm. To test the sensors we created a sensor event method that receives continuous data from the sensors and then displays it on a windows form. The IR sensor works by sending a beam of light continuously, this is then reflected back when it encounters an object. The light that is reflected back is detected by the IR sensor detector and this creates a triangle between the IR sensor, the object and the IR Sensor's detector. The angle of this triangle is then measured and used to calculate the distance to the object [7].

The Dr Robot manual states that if the data returned from the IR sensors is >=3446 then the object is 0-8cm away, if it returns a value between 5885 to 2446 then the object is 80-8cm away and if it is <=595 then the object is outside the range of detection. The data that is returned is nonlinear; it is therefore necessary to calculate the actual distance in cm using an Analogue to Digital conversion method created by Dr Robot. Displaying the IR data in meters instead of the raw value is more user friendly and easily understood.

We tested the sensors and noticed that the accuracy varied. For example, an object was placed approximately 20cm from the robot's IR7 sensor, which is located on the left side of the robot. The value returned was 3545, which was then converted to the more meaningful value of 9cm. An 11cm difference between the actual distance of an object and the distance reported could cause problems if the robots were operating in close proximity to one another. The IR sensors have proven to be less accurate than the Ultrasonic, which tends to give consistent readings. Further testing is required to ascertain whether different lighting and reflective objects give more or less accurate results. Due to this inaccuracy however, we plan to use either solely the Ultrasonic or the Ultrasonic in combination with the IR. The purpose of the sensors will be to enable the robots to avoid collisions with walls, objects and other operating robots.

### C. Human

The X80-H has 2 human sensors [8] that are situated on the base unit and facing upwards. The sensors can detect humans at a distance of 5m and human movement to a distance of 1.5m [9]. The human sensors are passive infrared sensors (PIR) that detect levels of infrared radiation emitted by humans. Unlike the 8 Infrared sensors on the X80-H, the human sensors do not emit an infrared beam.
The return value for the sensors is between 0 and 4095. When there is no human present, the left and right sensor data fluctuates between approximately 2000-2040. When a human is present the data drops to between 1700-2000, the data is constantly changing and it is difficult to obtain a figure that stays the same when a human is not moving.

The Human Motion sensor detects movement, like the Human Alarm it returns a figure of approximately 2000-2040 when no human is detected and then fluctuates rapidly between 0-4095 when there is movement. Testing has seen results of 1700-2400 when moving in front of the sensors. The purpose of the human motion sensors is to track which direction the human is moving, one way of doing this would be to compare the before and after readings from both sensors, create a threshold range representing 'no movement'. It would then be possible to determine if movement had occurred in front of either sensor by comparing the new value from the human motion sensors with the old value. If someone was to walk in front of both sensors, from the left side of the robot to the right side, the left human motion sensor should detect this first, making it possible to deduce the intended direction.

To detect then whether a human is present, the data from the Human Alarm sensors should be compared to the threshold range. To test the sensors, we created a test that checked whether the left human alarm sensor data was within the range of 1900-2200, if it was then the label displayed "No Humans". If the sensor value dropped below or above this range the label changed to "Humans Detected", the sensor data did change as expected when a human was in front of the sensor. The sensors are limited in that they cannot differentiate between people and cannot tell whether there is more than one person or how close they are to the sensor.

### D. GPS

Our version of the X80-H robot is equipped with an indoor GPS sensor that uses landmarks placed on the ceiling to calculate the position of the robot. The GPS sensor in the X80-H is not true GPS, it is a combination of an IR sensor and camera that checks the pattern on a landmark, it then creates an image from the pattern and analyses it to determine the robot's angle and position. The ideal ceiling placement is 2 meters apart; this ensures that there is no dead zone.[10]

### E. Camera

The wireless camera feed will sometimes become distorted with colored lines or a black screen. We have however been able to use it to take a still image and then perform image processing on the image to check for certain colors. This was very dependent on the lighting available and success was varied. For future experiments involving the detection of color coded floor areas, optimum lighting conditions will need to be established in order to produce consistent results. When connecting to two robots from within the same application, the camera feed from one robot will sometimes appear twice instead of the two different video feeds showing. This problem did not happen when we created two applications running on the same PC, one for each robot and using separate Gateway programs.

We plan to use the camera to detect different colored floor/terrains, for this we will use EMGU CV which is C# wrapper for OpenCV. Each robot will periodically take a snapshot of the floor in front of it and then perform color processing to check whether it is capable of continuing on the floor/terrain. Each robot will have its own role and will be aware of which terrain it cannot cross; the floor will be divided into different colored regions each representing a difficulty level. Figure 1 is an application we created that used image processing to detect the colour orange from a snapshot of the robot's camera and display the results in an ImageBox. The frst image shows the camera feed, the 2nd image is the snapshot of the feed, the 3rd image is the black and white processed image with the orange section of the image highlighted as white.

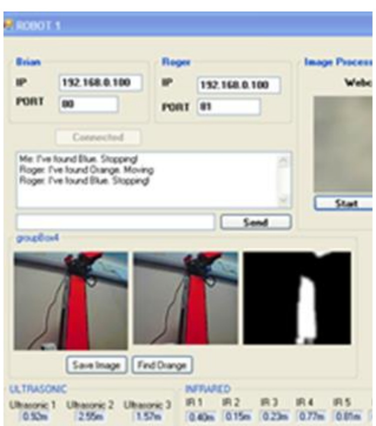


Figure 1. GUI showing a processed image with orange sections highlighted in white.

As part of our research we also experimented with using webcams, one for each robot, connected to a PC. We used the webcams in conjunction with EMGU CV and checked every frame for the presence of either orange or blue. If orange pixels were detected, the robot would move forward and then only stop when the webcam detected blue pixels. We were able to control two robots using two webcams and an orange and blue object, the robots would then exchange information about which colours they had found. Using UDP/IP, Robot1's applicaton would communicate with Robot2's application and exchange messages indicating that it had found a colour.

Image processing can be used to enhance the capabilities of a mobile robot, by providing the robot with an additional sense. In future we may consider using IP cameras that could be placed directly on the robot and test whether their camera is more reliable than the built in camera. Another possibility is to position IP cameras in the environment to act as mobile video sensors that a robot could theoretically have deposited. This would then act as an external camera that the robot's PC application would receive images from and provide more context than just the robot's sensors and camera.

## III. RESEARCH BACKGROUND

The purpose of this research is to investigate how mobile robots could use Autonomic Computing concepts to improve collaboration. To collaborate effectively, robots need to be self-aware and aware of their external environment. Each mobile robot operating in a swarm represents an Autonomic Element (AE) that is capable of self-management. A swarm of mobile robots are then part of a larger Autonomic System. The relationship between these AEs, how they interact and share information is the focus of this research.

When we talk about a swarm of robots or swarming it is generally understood to refer to a large group of entities that behave in a similar way and influence each other's behavior by their actions. The flocking behavior displayed by birds is achieved by each member following relatively simple rules. The Boids [11] [12] program was created to simulate flocking, it consists of multiple entities each following simple rules that enable them to avoid collisions and to follow the general direction and position of the other members of the flock.

Swarms function as a P2P system; there is no central controller that is coordinating the overall behavior or goal. Each member of the swarm is operating on simple rules and is not aware of the overall objective or purpose. For large systems it is perhaps more feasible for each entity to communicate via swarming as opposed to direct intelligent communication. To create an intelligent large swarm some form of hierarchy needs to be implemented so that communication and co-operation can occur without damaging the efficiency of the swarm.

The NASA Autonomous Nano-Technology Swarm (ANTS) project consists of several missions that will require significant advancements to be made in the area of Autonomic Computing [13]. One such mission is the Prospecting Asteroid Mission (PAM), which would involve sending a swarm of a 1000 small craft to explore and map an asteroid belt [14]. To reduce the amount of communication traffic that would occur, each craft would belong to a small cluster made up of Workers, Messengers and a Ruler [15]. Each small team within the swarm must adhere to the Self-CHOP paradigm so that the swarm as a whole can function efficiently [13][16].

Another NASA mission proposal was the ARES (Aerial Regional-scale Environmental Survey of Mars) [17], it would have involved the deployment of a glider plane on Mars. The MAVEN [18] mission was selected instead of ARES, MAVEN is an orbiter designed to study the Martian atmosphere, it was launched in 2013. The ARES plane would have flown a mile above the surface for 1 hour, with the purpose of detecting the source of methane gas that has been detected by satellites orbiting Mars [19]. Using a plane would have allowed information to be gathered over greater distances than the rovers are able to travel, it would also have been able to scout for possible human landing locations [17] [19]. The time delay from Earth renders teleoperation impossible; the software on the glider would have had to be pre-programmed with instructions. Future missions using a plane could benefit from autonomous software that enables it to adapt to its environment. If a future mission was designed so that a propulsion system was included in the plane, then Autonomic software would be vital to make sure that errors occurring during landing and takeoff could be corrected in real time.

The thin atmosphere makes it more difficult to sustain flight on Mars without a propulsion system. A more practical option would be to target Saturn's moon Titan; it is the only other planetary body in the Solar System aside from Earth that has an atmosphere. Due to the low gravity, sustaining flight would be much easier than it is on both Earth and Mars. The AVIATR (Aerial Vehicle for In situ and Airborne Titan Reconnaissance) [20] is a proposed 2020s mission that would operate for 1 year over Titan. The atmosphere on Titan is so thick that it is difficult to photograph the surface from orbit. As part of its mission, the AVIATR plane would map the surface and identify potential landing sites for future missions [20]. Due to the length of time that the plane would be in operation, it would need a high degree of self-management.

Our work will include an architecture that allows robots to collaborate to carry out a task such as searching for an object within a room. To carry out the research we will use four Dr Robot X80-H robots. In this case the number of entities is limited to four as opposed to an ever-changing large swarm. The research will focus on the collaboration between robots that are operating within a small cluster. Adaptation via role switching as a response to their current internal or external environment will also be explored. Future research may draw inspiration from the NASA PAM project and involve the exploration of how different clusters of robots collaborate within a larger swarm.

Internal self-configuration of the robot's system is important when it is faced with a dynamic external environment; it needs to be able to change its behavior to cope with the current situation. The idea of having different roles that can be activated depending on different external or internal situations will also be explored.

A robot that has been tasked with searching the environment could become damaged making it unable to move. To be truly autonomic it must recognize that it has been damaged; the next step would then be for it to notify other members of the group that it has been compromised. Changing its internal profile and channeling its efforts to processing information gathered by other members of the group would mean it was not completely redundant. This could involve it switching roles with another robot that had been tasked with keeping a global map and position of each robot. Another idea is to have the robots assess the area that was being searched by the damaged robot and rank its importance and likely impact on the global goal being achieved.

Self-management is not a new area for mobile robotics; autonomous behavior is the ultimate goal for swarming applications and other bio-inspired research. The difference between other self-management techniques and Autonomic computing is the MAPE-K control loop [1] [21]. In a mobile robot, an Autonomic Manager is responsible for managing the software and hardware components. In a swarm of autonomic mobile robots, each has an AM that can analyze the situation at hand, plan an appropriate response and store information that has been gathered. Mobile robotics is still a discipline which lacks standardization; most research focuses on solving SLAM or other standalone problems. Autonomic Computing is an attempt to create an agreed self-management architecture that can be standardized and built upon.

## IV. COLLABORATION

A self-managing system consisting of several independent Autonomic Elements is only truly autonomic if those Elements can co-operate to resolve a problem that has arisen.

In [22] a P2P system model is presented which allows a group of agents to contact each other directly or indirectly via a central controller agent. The agents send a receipt acknowledging that they received the message. Using a central controller to coordinate network traffic is a good idea when there are many agents in the system. This is a similar idea to a Super-Peer model were there are many Super-Peers within the system, each responsible for a cluster of agents or in our case robots [23][24].

Communication between robots that adhere to the MAPE-K loop could allow for information to be distributed and accessible by all in a swarm. In [21] they propose a distributed knowledge system that would allow each robot to compare its local goals with the goals of the group, enabling it to make more informed decisions. As the distributed knowledge was always being broadcast and updated, each robot would be aware of where and what its neighbors are doing [21].

In contrast to a distributed knowledge model is the idea that robots operating within a swarm only exchange data when necessary, for instance to collaborate on an immediate task. The MAST [25] project consists of a swarm of robots that map an indoor office environment using a video camera to detect doors and window, and a laser scanner to measure the distance to walls. Each robot creates a map and takes a note of its own location within the map, when 2 robots are within close proximity to each other, they exchange map data. The robots operate in a P2P manner, there is no Ruler robot, all robots are equal and each is capable of recruiting others to help them map an area more efficiently. They can send messages to each other e.g. “I’m going to the left if you go to the right” this results in a faster mapping of an area and less duplication [25].

Collaboration does not just involve communication strategies, it is also important that entities can adapt to a situation and perhaps change how members of a swarm behave in order to benefit the group as a whole. The notion of role switching, being able to change the focus and abilities of a robot when presented with different scenarios could prove useful if some robots were damaged or lost. Dynamically assigning different roles to a set of homogeneous robots has been explored by the SWITCH project [26]. Developed for the Robocup, the SWITCH robots could change their role from Striker to Defender in response to how the game is progressing; each role has its own goals and strategies. To determine which role a robot is best suited to, a number of factors such as ‘Distance to ball’ are checked periodically by the robot and given a weighted value. The robot then checks whether it should be a Defender or Striker based on this value, it also transmits its position and role to the other robots. Each robot maintains a global model of its teammates position and distance from the ball [26].

Indirect communication can be achieved by changing an environment in a way that means something to the others operating in that environment. This is known as Stigmergy; it is seen in nature and is the basis of much bio-inspired research. An idea presented in [27] is that of using pheromones to communicate which areas have been mapped. The robots leave a pheromone trail from start to finish, if another robot detects this trail it knows not to proceed.

## V. PROPOSED RESEARCH

For the research we will use 2-4 robots and have them operate in a room with a color-coded floor layout. Each color will represent a level of difficulty and each robot will be given a role that dictates which terrain they cannot traverse. The X80-H camera is embedded in the right eye, this will need to be angled toward the floor, the terrain color checking will occur at timed intervals e.g. every 30 seconds.

Their task is to find a colored object while avoiding collisions and collaborating when certain situations arise that may require them to switch roles and position. Other scenarios could involve a robot simulating damage to its drive system and notifying the others that it cannot continue to search; this would require the cluster to reassess priorities and assign a robot to search the damaged robot's terrain.

In Figure 2, an overview of the system is displayed; each robot will have an Autonomic Manager application running on a host PC. The physical robot represents the Managed Component of the Autonomic Element. The Robot Autonomic Manager applications will be able communicate and exchange information via UDP/IP. We have already done preliminary development using UDP/IP but may decide to use TCP/IP as messages in UDP/IP are not acknowledged and may not be received in the correct order.
A Mapping Autonomic Element application also runs on the PC, its keeps track of the position of all of the robots within the cluster and which role and ability each currently possesses. If a robot encounters terrain that it cannot cross, it would notify the Mapping AM which would then notify a robot that is able to operate on that terrain. The robots would then change positions by querying the Map AM for the co-ordinates of the other and then move toward that location.

In [28] a similar idea is proposed, different arenas were created each with varying levels of difficulty. Objects within the environment were painted either red or yellow with red representing rescue victims.

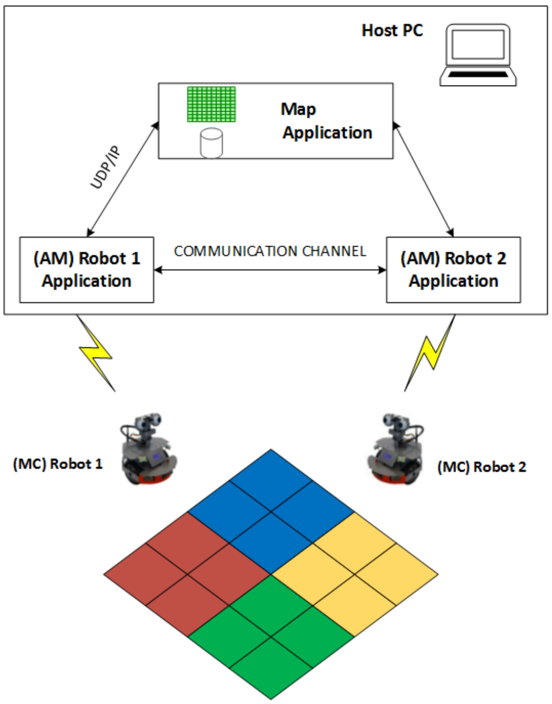


Figure 2. System Overview

Development of the system will be carried out in Visual Studio using C#, for image processing we are using EMGU CV. For communication between different AM applications running on the same PC, we will use UDP/IP or TCP/IP. The sensors will be used for collision detection; at this point the Ultrasonic sensors have proven to be more reliable and accurate than the Infrared. The GPS Stargazer system will be used to locate each robot in the environment. We plan to use the Indoor Stargazer GPS system to keep track of the robots; however this has proven a challenge due to a lack of documentation. The plan is to use the Mapping application to display a grid showing where each robot is based on the information from the indoor GPS.

If this is not possible, we have considered an alternative whereby each robot moves 30cm at a time and then pauses; the robot sends its movements to the Mapping application which then moves a Robot Icon on a grid. Each cell in the grid would represent a 30cm by 30 cm square of the floor space. If a robot wished to move to another robot's location, it would query the Mapping application which would then calculate the sequence of manoeuvres and relay them e.g. Move forward 90cm, Turn Left 90 degrees. A similar mapping idea based on odometry wheel rotation and fixed size squares is presented in [29] and allows a robot to map unknown environments. The robot generates maps by using its sensors; the maps include structures within the data center. They used an iCreate robot and a webcam to view the floor tiles, the software uses the floor tiles as a marker and moves the robot 1 tile at a time helping it to keep track of where it is. The robot takes sensor readings of the room each time it moves and also takes an image of the tile directly ahead of it; a software visualization system then uses this information to construct a graphical view of the room.

Figure 3 is a preliminary design showing the Mapping Autonomic Element and Robot Autonomic Element exchanging data via a Communications Channel. The Mapping Autonomic Element consists of an Autonomic Manager that monitors the External Environment for data from the Robot Autonomic Elements. It uses this data to keep track of the position of each robot and displays an icon of each robot on a grid. The Communications Channel can also be used by a Robot AM to request information such as the location of other robots. The Mapping Managed Component (MC) consists of a Data Storage module and a Planner/Analyzer module, the purpose of the latter is to suggest paths and organize the role switching.

The Robot Autonomic Element differs to the structure of the Mapping AE; its MC is the hardware of the robot, the AM checks the data sent back from the robot and uses image-processing algorithms to detect whether the target object has been found. The sensor data is used for collision detection.

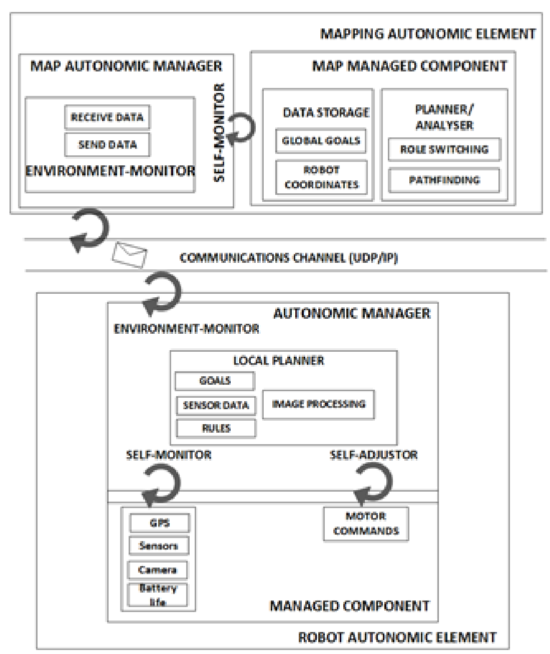


Figure 3. Mapping AE and a Robot AE

## VI. CONCLUSION

The area of mobile robotics is a vast research area; there is also a great deal of work being carried out in Autonomic Computing, however there is a lack of overlap between the two areas. This project will seek to apply the Autonomic Computing MAPE-K control loop to a cluster of mobile robots. This paper has explained what sensors will be used for the research and what their limitations are. An overview of the proposed system has been presented and explained. Future research will focus on the practical application of the system and the design of a set of collaboration rules and roles for Autonomic Elements.

This research project aims to add to the fairly new field of Autonomic Computing. Autonomic Computing could greatly benefit future space missions and it is therefore fitting to use robots to practically demonstrate this. Internal self-management of each mobile robot will follow the MAPE-K control loop structure. Our research will include the development of communication rules for heterogeneous Autonomic Elements. These rules will enable data exchange and collaboration to take place. Applying Autonomic Computing to robotics is a relatively unexplored research area; our aim is to contribute ideas and to generate further research interest in this area.

## REFERENCES


[1] IBM, "An Architectural Blueprint for Autonomic Computing," IBM, 2003.

[2] R. Sterritt, M. Parashar, H. Tianfield, and R. Unland, "A Concise Introduction to Autonomic Computing," *Advanced Engineering Informatics*, vol. 19, no. 3, pp. 181–187, Jul. 2005.

[3] A. G. Ganek and T. A. Corbi, "The Dawning of the Autonomic Era," *IBM Systems Journal*, vol. 42, no. 1, pp. 5–18, 2003.

[4] R. Sterritt, "Towards Autonomic Computing : Effective Event Management," in *Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02)*, 2003, pp. 40–47.

[5] R. Sterritt and D. Bustard, "Towards an autonomic computing environment," in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, 2003, pp. 694–698.

[6] D. Robot, "DUR5200 Ultrasonic Range Sensor Module User Manual," 2004. [Online]. Available: http://www.drrobot.com/products/item_downloads/DUR5200_1.pdf.

[7] "Infrared vs Ultrasonic - What You Should Know." [Online]. Available: http://www.societyofrobots.com/member_tutorials/book/export/html/71. [Accessed: 17-May-2014].

[8] Dr Robot, "DHM5150 Human Motion Sensor Module User Manual," 2005. [Online]. Available: http://www.drrobot.com/products/item_downloads/DHM5150_1.pdf.

[9] D. Iercan, E. Voisan, L. Fedorovici, and F. Dragan, "nRobotic: A platform for managing robots with different capabilities," in *2009 5th International Symposium on Applied Computational Intelligence and Informatics*, 2009, pp. 187–192.

[10] "Hagisonic StarGazer Robot Localization System." [Online]. Available: http://www.robotshop.com/en/hagisonic-stargazer-localization-system.html. [Accessed: 16-Apr-2014].

[11] C. Reynolds, "Boids," 1995. [Online]. Available: http://www.red3d.com/cwr/boids/. [Accessed: 17-May-2014].

[12] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model." [Online]. Available: http://www.macs.hw.ac.uk/~dwcorne/Teaching/Craig Reynolds Flocks, Herds, and Schools A Distributed Behavioral Model.htm. [Accessed: 17-May-2014].

[13] R. Sterritt, C. Rouff, J. Rash, W. Truszkowski, and M. Hinchey, "Self- * Properties in NASA Missions," in *4th International Workshop on System/Software Architectures (IWSSA'05) in Proc. 2005 International Conference on Software Engineering Research and Practice (SERP'05)*, 2005.

[14] R. Sterritt and M. Hinchey, "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions," in *Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, 2005, pp. 506–511.

[15] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, "Swarm Technology at NASA: Building Resilient Systems," *IEEE IT Pro*, vol. 14, no. 2, pp. 36–42, 2012.

[16] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "NASA's Swarm Missions: The challenge of Building Autonomous Software," *IEEE IT Pro*, vol. 6, no. 5, pp. 47–52, 2004.

[17] NASA, "ARES Aerial Regional-scale Environmental Survey of Mars," 2011. [Online]. Available: http://marsairplane.larc.nasa.gov. [Accessed: 28-May-2014].

[18] NASA, "MAVEN Mars Atmosphere and Volatile Evolution," 2014. [Online]. Available: http://www.nasa.gov/mission_pages/maven/main/index.html#.U4Xxg165Mds. [Accessed: 28-May-2014].

[19] "ARES An Airplane over Mars," 2008. [Online]. Available: https://www.youtube.com/watch?v=8YutbpJuFiI. [Accessed: 28-May-2014].

[20] J. W. Barnes, L. Lemke, R. Foch, C. P. McKay, R. A. Beyer, J. Radebaugh, D. H. Atkinson, R. D. Lorenz, S. Le Mouélic, S. Rodriguez, J. Gundlach, F. Giannini, S. Bain, F. M. Flasar, T. Hurford, C. M. Anderson, J. Merrison, M. Ádámkovics, S. a. Kattenhorn, J. Mitchell, D. M. Burr, A. Colaprete, E. Schaller, a. J. Friedson, K. S. Edgett, A. Coradini, A. Adriani, K. M. Sayanagi, M. J. Malaska, D. Morabito, and K. Reh, "AVIATR—Aerial Vehicle for In-situ and Airborne Titan Reconnaissance," *Experimental Astronomy*, vol. 33, no. 1, pp. 55–127, Dec. 2011.

[21] J. P. Barraca, R. Sadeghi, and R. L. Aguiar, "Collaborative Relaying Strategies in Autonomic Management of Mobile Robotics," *Springer*, 2013. [Online]. Available: http://download.springer.com/static/pdf/770/art%3A10.1007%2Fs11277-013-1104-1.pdf?auth66=1384953190_e0c035b50307361094641efcc6bb2ae9&ext=.pdf.

[22] T. Enokido and M. Takizawa, "Group Communication Protocol for Autonomic Computing Tomoya Enokido," in *Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, 2005.

[23] A. Saini, "Super-Peer Architectures for Distributed Computing." [Online]. Available: http://www.fiorano.com/whitepapers/superpeer.pdf. [Accessed: 19-May-2014].

[24] B. Beverly Yang and H. Garcia-Molina, "Designing a Super-Peer Network," in *Proceedings of the 19th International Conference on Data Engineering*, 2003, pp. 49–60.

[25] B. Coxworth, "Autonomous robots could collaborate on mapping hazardous environments," 2011. [Online]. Available: http://www.gizmag.com/mast-collaborative-robot-mapping-project/18655/. [Accessed: 16-Apr-2014].

[26] C. E. Aguero, V. Matellan, J. M. Canas, and V. M. Gomez, "SWITCH ! Dynamic Roles Exchange Among Cooperative Robots," in *Proceedings of the 2nd International Workshop on Multi-Agent Robotic Systems - MARS 2006 INSTICC*, 2006, pp. 99–105.

[27] N. Capodieci and G. Cabri, "Collaboration in Swarm Robotics : a Visual Communication Approach," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 195–202.

[28] A. Chiou and C. Wynn, "Urban Search and Rescue Robots in Test Arenas: Scaled Modeling of Disasters to Test Intelligent Robot Prototyping," *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pp. 200–205, 2009.

[29] J. Lenchner, C. Isci, J. O. Kephart, C. Mansley, J. Connell, and S. McIntosh, "Towards Data Center Self-Diagnosis Using a Mobile Robot," in *Proceedings of the 8th ACM International Conference on Autonomic Computing - ICAC '11*, 2011, pp. 81–90.